

公開実用平成 3-119217

⑩ 日本国特許庁(JP)

⑪ 実用新案出願公開

⑫ 公開実用新案公報(U) 平3-119217

⑬ Int. Cl.⁵

識別記号

庁内整理番号

⑭ 公開 平成3年(1991)12月9日

G 06 F 1/14

7459-5B G 06 F 1/04 351 A

審査請求 未請求 請求項の数 1 (全 頁)

⑮ 考案の名称 システム時刻自動修正装置

⑯ 実 願 平2-27056

⑰ 出 願 平2(1990)3月16日

⑱ 考 案 者 関 豊 東京都武蔵野市中町2丁目9番32号 横河電機株式会社内
⑲ 出 願 人 横河電機株式会社 東京都武蔵野市中町2丁目9番32号
⑳ 代 理 人 弁理士 小沢 信助

明 細 書

1. 考案の名称

システム時刻自動修正装置

2. 実用新案登録請求の範囲

計算機制御システムにおけるシステム時刻の自動修正装置であって、

クロック割込み手段と、

設定現在時刻を受け、この時刻とシステム現在時刻から時刻修正量を算出する時刻修正量演算手段と、

この時刻修正量演算手段で演算された時刻修正量を前記クロック割込み手段による割込みを受けて、システム時刻を n 回（ n は任意の整数）に1回の割合で徐々に設定現在時刻に修正する時刻修正手段と、

この時刻修正手段によるシステム時刻の修正が完了した後、所定の補正周期に1回器差補正の為の時刻補正を行う時刻補正手段とを設け、

時間差の修正と器差による遅れ／進みの補正を定周期処理や定刻処理に影響を与えないで行える

ようにしたことを特徴とするシステム時刻自動修正装置。

3. 考案の詳細な説明

<産業上の利用分野>

本考案は、計算機制御システムにおけるシステム時刻の自動修正装置に関し、更に詳しくは、システム時刻の修正を通常の制御動作に影響を与えないで行えるシステム時刻自動修正装置に関する。

<従来技術>

計算機システムでは、従来から内部クロックを分周することで、クロック割込み機能を実現している。クロック割込みがあると、ソフトウェアはこのクロック割込みを伝達して、システム時刻をカウントアップさせたり、又は、タイマ管理機能を実現している。

この様なシステムにおいては、長期の連続運転に伴ない、標準時間からの積算誤差が生ずることが避けられない。

このため、従来よりこの様な標準時間からの積算誤差を修正するために、通信回線を経由してホ

ストコンピュータや手動時刻エントリー手段により、時刻合わせを行っていた。

< 考案が解決しようとする課題 >

しかしながら、このような時刻修正の手法は、次のような問題点があった。

(a) 時刻変更によって、定時処理がその前後において二重に起動されたり、起動の抜けが発生することがある。

(b) 定周期処理の起動間隔や起動回数が乱れることにより、制御動作に影響がでることがある。

本考案は、これらの点に鑑みてなされたもので、時刻合わせのイベントがシステムに生じても、時刻修正処理をクロック割込み処理の中で、修正を段階的に時間をかけて実施することにより、通常の動作に影響を与えないで、時刻修正がおこなえるシステム時刻自動修正装置を提供することを目的とする。

< 課題を解決するための手段 >

この様な目的を達成する本考案は、

計算機制御システムにおけるシステム時刻の自

動修正装置であって、

クロック割込み手段と、

設定現在時刻を受け、この時刻とシステム現在時刻から時刻修正量を算出する時刻修正量演算手段と、

この時刻修正量演算手段で演算された時刻修正量を前記クロック割込み手段による割込みを受けて、システム時刻を n 回（ n は任意の整数）に1回の割合で徐々に設定現在時刻に修正する時刻修正手段と、

この時刻修正手段によるシステム時刻の修正が完了した後、所定の補正周期に1回器差補正のための時刻補正を行う時刻補正手段とを設けて構成される。

<作用>

時刻合わせのイベントがシステムに生ずると、時刻修正手段は、クロック割り込み処理の中で修正を段階的に時間をかけて実施する。

これにより、時間差の修正と器差による遅れ／進みの補正を定周期処理や定刻処理に影響を与え

ないでシステム・タイマ処理を行うことが可能になる。

< 実施例 >

以下図面を用いて、本考案の実施例を詳細に説明する。

第1図は、本考案が適用されるシステムの一例を示す構成ブロック図である。図において、1はマイクロプロセッサで、ソフトウェアにより時刻修正量演算手段や、時刻修正手段、時刻補正手段の機能を行う。2はクロック制御インターフェース部で、水晶発振器からのクロックを分周する分周器21と、この分周器からのパルスを受けて割込みを発生する割込み発生回路22を含んで構成されている。3は処理プログラムを収納したROM、4は電池によりバックアップされた不揮発メモリ、5はプログラムワークメモリとしてのRAMで、これらは内部バスBSを介して結ばれている。

6は内部バスBSに接続される時刻エントリ手段を総括的に示したもので、61はマンマシン



・インターフェース回路、62はCRT及びキーボード等である。71はデータ通信インターフェース回路、72はデータ通信インターフェース回路71に通信回線を介して接続された他のホストコンピュータあるいは標準時刻発生器である。

この時刻エントリー手段6は、時刻を修正するための補助手段であって、外部からシステムに対して修正すべき設定時刻を通知する機能をサポートするもので、他の手段で実現してもよい。

このように構成される装置による動作を説明すれば、以下の通りである。

第2図は、時刻修正イベントをシステムが受信した場合の、強制時刻修正イベント処理及び自動時刻修正イベント処理の手順を示すフローチャートである。ここでのイベントの発生は、マンマシン・インターフェース部61を介して、キーボード等62から、あるいは通信回線を介して接続されている標準時刻発生器72（または標準として定めたホストコンピュータ）によって行われ、現在時刻が伝文として同時に伝達される。また、標

準（基準）とすべき時刻をいずれからか参照できるものとする。

また、不揮発メモリ 4 内には、処理の連続性を保つために、前回修正時刻、補正周期、時刻修正中カウンタ、現在システム時刻等の情報があって、これらが順次更新されるようになっているものとする。

強制時刻修正イベント処理の場合、その時刻修正が初回であれば、COUNTを「0」、補正周期 = 0 にセットし、時刻修正中カウンタの値を「0」にセットする。また時刻修正が初回でなければ、ステップ 2, 3 (ST 2, ST 3) をバイパスし、時刻修正中カウンタの値を「0」にセットする。その後、前回修正時刻を設定現在時刻に、また現在システム時刻を設定現在時刻にそれぞれ修正する (ST 5, ST 6)。

他ホストコンピュータあるいは標準時刻発生器 7 2 からの指示に基づく自動時刻修正イベント処理の場合、はじめに、修正量 = (設定現在時刻 - システム現在時刻) を演算し、修正量が自動修正

制限値を越えているか否かを判断する（ST7, 8）。ここで修正量が自動修正制限値より大きい場合は、ステップ1に分岐し、強制時刻修正イベント処理と同様な手順をとる。

ステップ8において、修正量が自動修正制限値より小さい場合は、ステップ9で、以下の計算を行う。

$$x = (\text{設定現在時刻} - \text{前回修正時刻})$$

$$y = (\text{現在システム時刻} - \text{設定現在時刻})$$

ここで、現在システム時刻と設定現在時刻との大きさを比較し（ST10）、両者が同じの場合、すなわち $y = 0$ の場合は、補正周期 = 0 とし（ST12）、現在システム時刻と設定現在時刻とが同じでない場合、すなわち $y \neq 0$ の場合は、補正周期を次の式に従って演算する（ST11）。

$$\text{補正周期} = \text{NINT} (x / (y * t1))$$

続いて、時刻修正中カウンタの値を

$$\text{NINT} (y / t1)$$

に設定し、COUNT = 0 とし、続いて前回修正時刻 = 設定現在時刻にして終了する（ST13）

～15)。

第3図は、クロック制御インターフェース部2からのクロック割り込み処理による時刻の修正、補正の手順を示すフローチャートである。このフローは、設定された一定周期 t_1 (mS) でクロック割り込みが発生して起動されるものである。

まず、時刻修正中カウンタが「0」か否かを判断し、「0」であれば時刻補正処理を、「0」でなければ、時刻修正処理を行う。ここで時刻修正処理の手順は、第4図のフローチャートに、また、時刻補正処理の手順は、第5図のフローチャートにそれぞれ示す。

これらの各処理により、今回のクロック割り込みで時刻の修正（進ませるか遅らせるか）をする／修正をしないかが判断され、その結果が、フラグに示される。すなわち、フラグが「0」の場合は、無修正、フラグが「1」の場合は、進ませる、フラグが「-1」の場合は、遅らせることがそれぞれ示される。

これらの処理を経た後、フラグの値を判断し

(S T 2 2)、フラグの値 F L A G ≥ 0 の場合は、システム時刻更新処理を行う (S T 2 3)。この時刻更新処理は、現在システム時刻を現在システム時刻 + t 1 にする処理である。

なお、ここでは、ソフトウェアタイマ・タイムアップ処理の後、F L A G $\neq 0$ の場合は、2 回分割込みサービスをするようにしている (S T 2 4 ~ 2 6)。

ステップ 2 2 でフラグの値 F L A G < 0 の場合は、今回の割込みは無視することで、時刻更新処理をバイパスして、時刻を遅らせる。

第 4 図のフローチャートに示す時刻修正処理は、自動時刻修正イベントを受信した後で、時刻差を一定の割合 (この例ではタイマ割込みインターバル t 1 に対して、1 0 t 1 に 1 回修正動作をする $\frac{1}{10}$ 割の割合) で、修正して行く為の動作である。

ここでは、時刻修正中カウンタが「0」を示している場合は、フラグを「0」に設定して (S T 4 0)、リターンすることになる。

時刻修正中カウンタが「0」でない場合、10回に1回の割合で、システム時刻の遅れ、進みを修正する。ここで、時刻修正中カウンタが「0」かそれより大きい場合は、時刻修正中カウンタの値を-1だけ減らし、フラグを「1」に設定し（ST36, ST37）、時刻修正中カウンタが「0」より小さい場合は、時刻修正中カウンタの値を+1だけ加え、フラグを「-1」に設定する（ST38, ST39）。

第5図のフローチャートに示す時刻補正処理は、定常状態でクロックの器差による遅れ／進みを自動的に補正する為の動作である。

ここでは、補正周期が「0」を示している場合は、フラグを「0」に設定して（ST58）、リターンすることになる。

補正周期が「0」でない場合、COUNT = (COUNT) - 1に設定し、COUNTが「0」寄り大きいかな否かを判断する（ST52, ST53）。COUNTが「0」より大きい場合は、ST58に行く。COUNTが「0」か又は「0」より

小さい場合は、補正周期になったとしてシステム時刻補正処理を行う。

ここでは、 $COUNT = (\text{補正周期})$ とし、次に（補正周期）が「0」より大きいか否かを判断し、それが「0」かそれより大きい場合は、フラグを「1」に設定し（ST56）、「0」より小さい場合は、フラグを「-1」に設定する（ST57）。

第6図は、以上の処理動作の全体の流れを示すタイムチャートである。

時刻修正イベントを受信すると、タイマ割込みにより10tに1回の割りで修正動作が行われ、システム時刻を $(\text{時間差} \times 10) / t$ かけて徐々に修正して行く。これにより時刻修正時に急激な時刻変更が行われることはないようにしている。

システム時刻の修正が完了すると、その後本来の定常状態に入るが、器差があるのでこの時刻補正を、補正周期ごとに1回実施する。

< 考案の効果 >

以上詳細に説明したように、本考案によれば、

標準時刻を外部参照することにより、システムの自動修正することができる。また、システム側で修正され、かつアプリケーション側から見ると、補正動作に特殊な処理を必要としないことから、リアルタイム・システム構築が容易になる。

また、時刻合わせのイベント発生タイミングは、システム内部で作成してもよく、この場合は、外部にある標準時刻を読み込むことで同一機能を実現できる。

4. 図面の簡単な説明

第1図は本考案が適用されるシステムの一例を示す構成ブロック図、

第2図は時刻修正イベントをシステムが受信した場合の強制時刻修正イベント処理及び自動時刻修正イベント処理の手順を示すフローチャート、

第3図はクロック制御インターフェース部からのクロック割り込み処理による時刻の修正、補正の手順を示すフローチャート、

第4図は時刻修正処理の手順を示すフローチャート、



第5図は時刻補正処理の手順を示すフローチャート、

第6図は処理動作の全体を示すタイムチャートである。

1…マイクロプロセッサ

2…クロック制御インターフェース部

3…ROM

4…不揮発メモリ

5…RAM

BS…内部バス

6…時刻エントリー手段

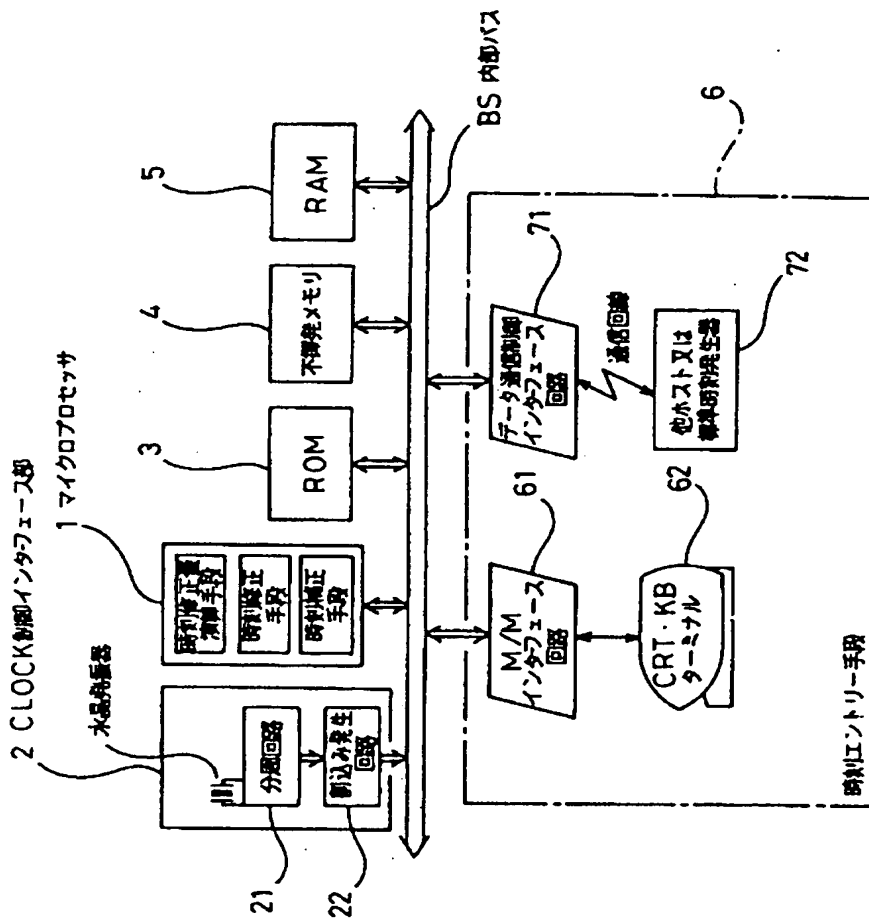
実用新案登録出願人

横河電機株式会社

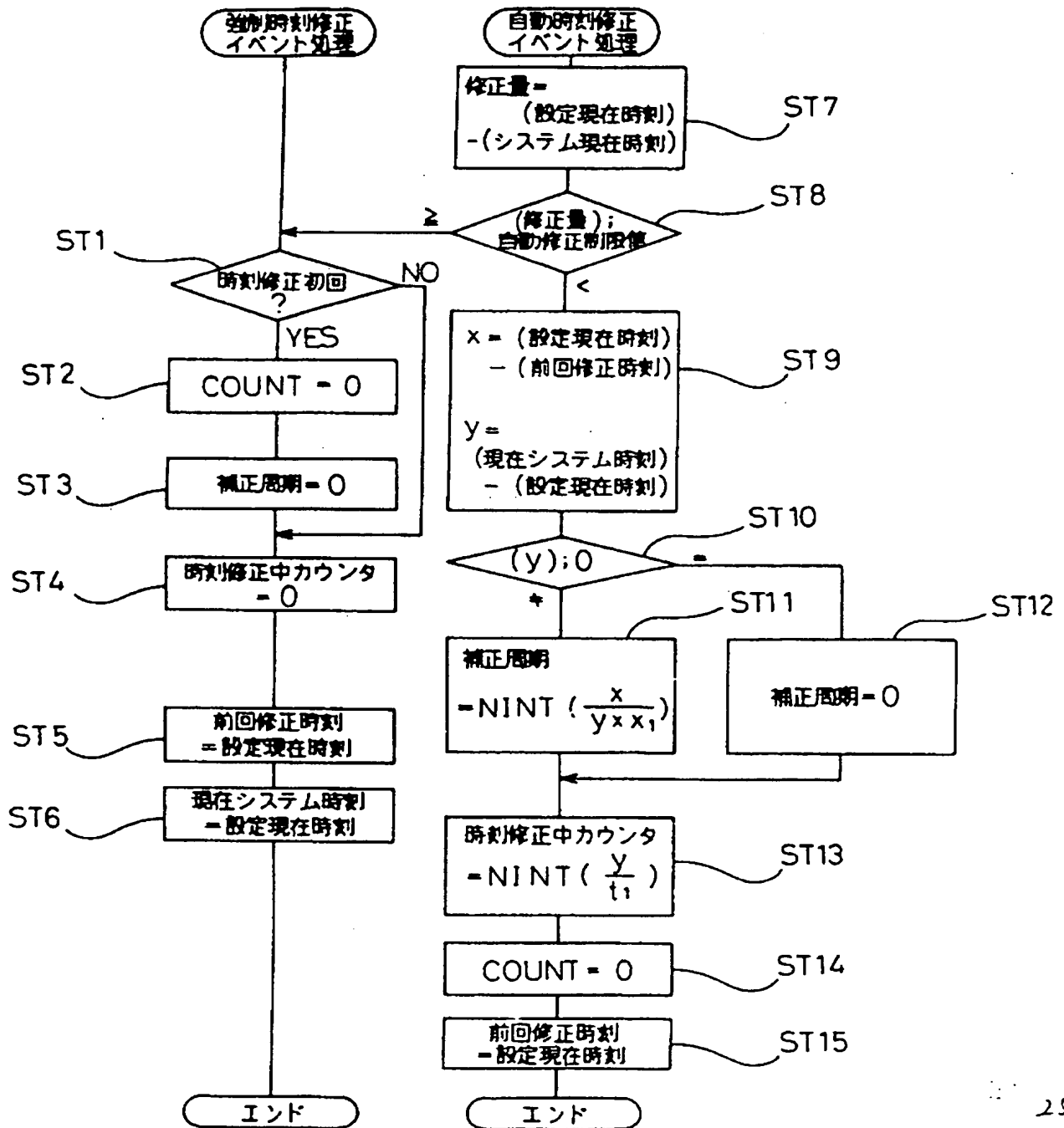
代理人 弁理士 小沢 信 助



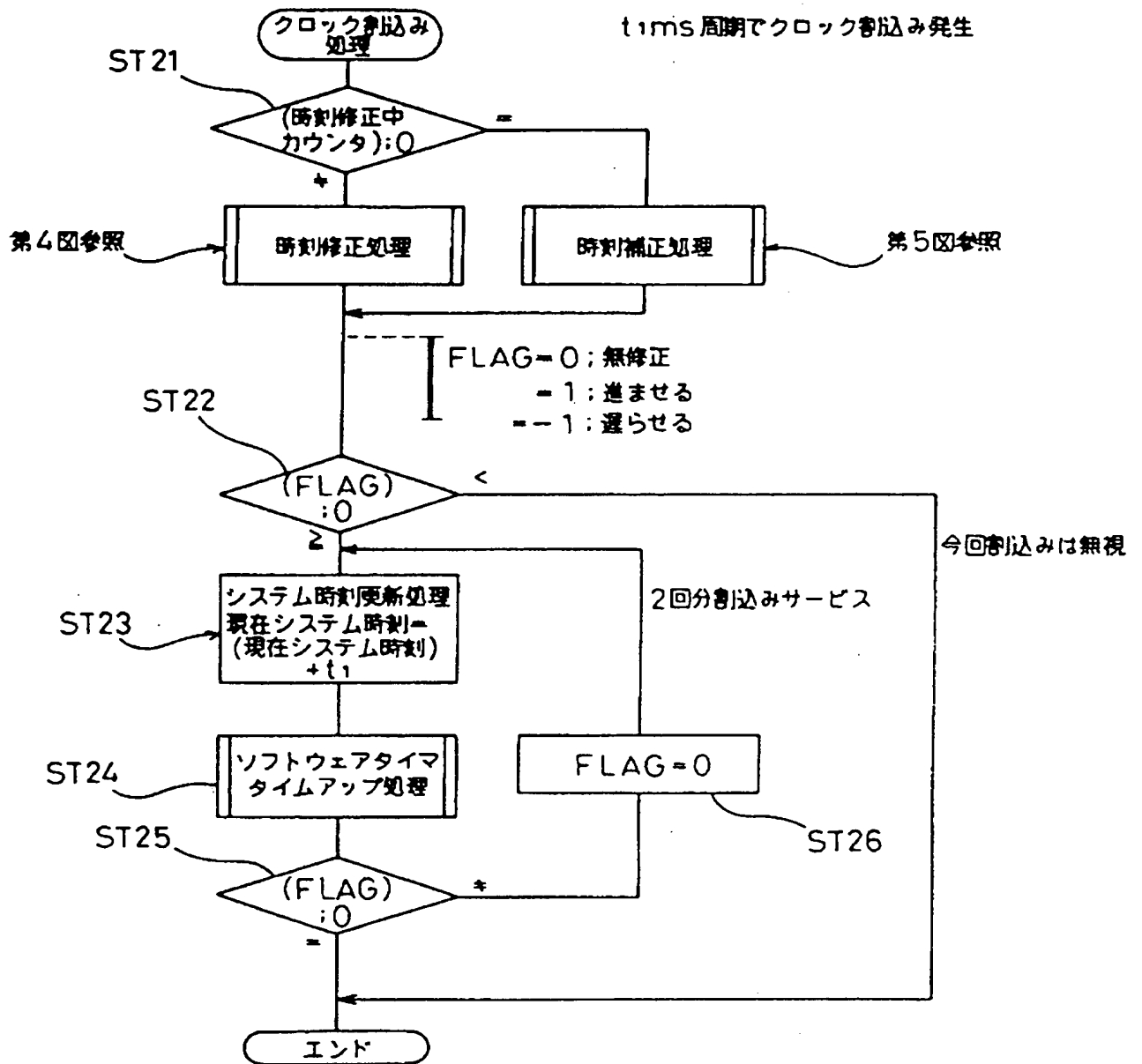
第 1 図

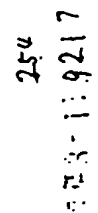


第 2 図

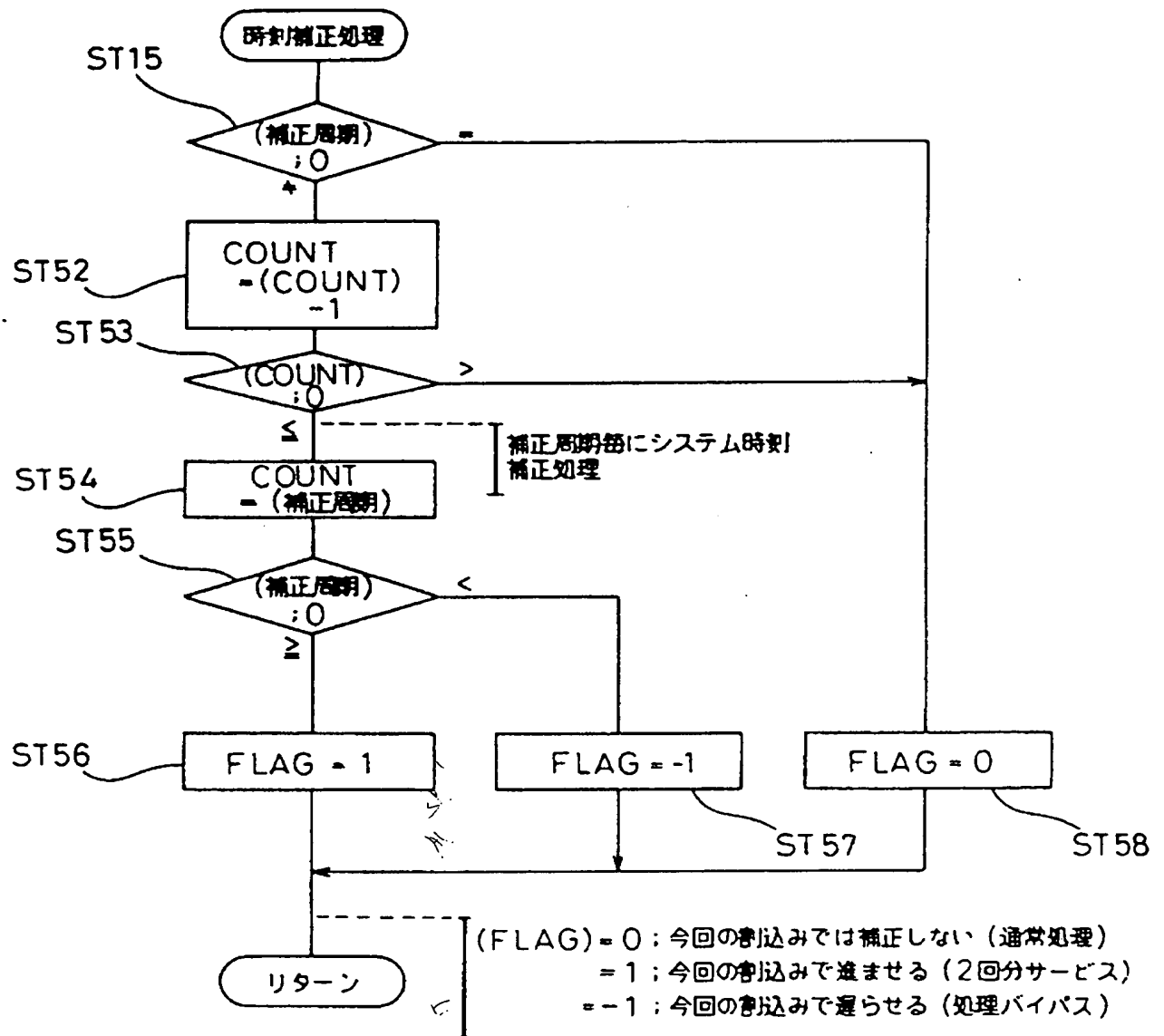


第 3 図



☒

第 5 図



第 6 図

